

TARTU ÜLIKOOL

MATEMAATIKA-INFORMAATIKATEADUSKOND

Arvutiteaduse instituut
Infotehnoloogia eriala

Julia Gritsuk

Tudengite õpiharjumuste jälgimise päevik
Bakalaureusetöö (6 EAP)

Juhendaja: Margus Niitsoo

Autor:

“.....” mai 2013

Juhendaja:

“.....” mai 2013

Lubada kaitsmisele

Professor:

“.....” mai 2013

Sisukord

Sissejuhatus	4
1. Ülevaade	5
1.1 Mis on tudengite õpiharjumuste jälgimise päevik?	5
1.2 Olemasolevad süsteemid.....	5
2. Süsteemi kirjeldus	7
2.1 Mittefunktsionaalsed nõuded	7
2.1.1 Kasutatavus	7
2.1.2 Käideldavus.....	7
2.1.3 Toetus	7
2.2 Funktsionaalsed nõuded.....	7
2.2.1 Sisselogimine	7
2.2.2 Ainele registreerimine	8
2.2.3 Ainele registreerimise tühistamine.....	9
2.2.4 Ajaaruandlus.....	9
2.2.5 Raporteerimata päevade vaatamine.....	10
2.2.6 Tulemuste vaatamine.....	11
2.2.7 Grupi loomine	11
2.2.8 Gruppi sisenemine.....	12
2.2.9 Grupist väljumine.....	12
2.2.10 Välja logimine	13
2.2.11 Abijuhendi avamine	13
2.2.12 Tudengi ja õppejõu eristamine	13
2.3 Graafikute ja diagrammide ehitamine.....	13
2.3.1 Diagrammide komplekt õppejõu jaoks	14
2.3.2 Diagrammide komplekt tudengi jaoks	16
2.4 Medalite saamine	18
3. Kasutatud tehnoloogiad	20
3.1 Veebirakenduse serveripoolne osa.....	20
3.1.1 Servletid	20
3.1.2 Sessiooni objekt, filter ja kuular.....	21
3.1.3 Autentimine.....	21
3.1.4 Kodeeringu filter	21

3.1.5 Andmebaas	21
3.2 Veebirakenduse kliendi poolne osa	23
3.2.1 <i>JavaScript</i> ja selle teegid.....	23
3.2.2 <i>HTML5</i> ja graafikute ehitamine	24
4. Kokkuvõte	25
Summary.....	26
Viited	27
Lisad	28
Lisa 1. Veebirakenduse failide sisu lühikirjeldus	28
Lisa 2. Rakenduse lähtekood	29

Sissejuhatus

Tudengielu on alati olnud väga sündmusterikas periood, millal õppimise ja võib-olla ka töö kõrval tuleb leida aega sõpradega kokkusaamiseks, spordiga tegelemiseks ja eraelu ehitamiseks. Noorena on jõudu kõikide tegevuste jaoks, kuid tihti tekib probleeme ajaplaneerimisega - lisaks sellele, et on vaja võimalikult hästi planeerida enda tegevuste aega, on tudengil vaja õppimisele jäetud aega jaotada erinevate ainete vahel. See tekitabki kõige rohkem raskusi. Ajaplaneerimise oskus võib tulla aastate möödumisel, kuid seda protsessi saab ka kiiremaks muuta. Üks viis selle tegemiseks on aidata inimesel visualiseerida inimestel enda kujunenud iganädalast ajaplaani, et ta saaks seda analüüsida ja vajaduse korral korrigeerida.

Antud töö autori eesmärgiks on disainida ja implementeerida veebikeskkond Tartu Ülikooli tudengite jaoks, kus õpilased saavad iga päev märkida, kui kaua ja mis õppimismeetodeid kasutades nad tegelesid valitud ainetega. Selline süsteem on vajalik tudengitele selleks, et kontrollida oma ajakulu erinevate ainete õppimiseks ja jälgida enda ja teiste tudengite õpiharjumuste kujunemist. Samuti võimaldab see õppejõul väga mugavalt saada ülevaadet tegeliku aine õppemahu ning tudengite õpistrateegiate kohta.

Töö esimene peatükk annab ülevaate valmiva süsteemi põhimõtetest ning sarnastest olemasolevatest lahendustest veebis. Teises peatükis vaadeldakse valminud veebikeskkonna võimalusi ning kuidas projekti eesmärgid olid saavutatud. Kolmanda peatüki teemaks on kasutatud tehnoloogiad ja nende valikute põhjused. Ning neljandaks ja viimaseks peatükiks on kokkuvõtte probleemi püstitamisest ja selle lahendusest ning edasiarendamise võimalustest. Lisadena on esitatud veebirakenduse failide sisu lühikirjeldus (vt. *Lisa 1*) ja rakenduse kogu lähtekood (vt. *Lisa 2*).

1. Ülevaade

1.1 Mis on tudengite õpiharjumuste jälgimise päevik?

Tudengite õpiharjumuste jälgimise päevik on selline süsteem, mis võimaldab tudengitel raporteerida enda ajakulu erinevatele ainetele. Ainete kohta ajaaruandluse tegemine on hea võimalus koguda informatsiooni selle kohta, kui palju aega kulub tudengitel ainetega tegelemisele. Edaspidi võivad seda informatsiooni kasutada nii tudengid ise, kui ka Tartu Ülikooli õppejõud.

Tihti juhtub, et tudengid ei oska planeerida enda aega ning jaotada seda mitmete vajalike tegevuste vahel. Sellest tulenevad ka probleemid ainete sooritamistega. Kui aga ajaaruandluste tegemisel panna kirja tõelisi tulemusi, siis päevikusüsteem saab heaks abivahendiks enda ajakulu kontrollimiseks ning tööplaanist õige pildi saamiseks. Sellisel juhul saab edaspidi analüüsida ja võrrelda enda ja teiste tudengite keskmist ajaplaani ning vajadusel muuta enda õpiharjumusi sobivamateks ja produktiivsemateks.

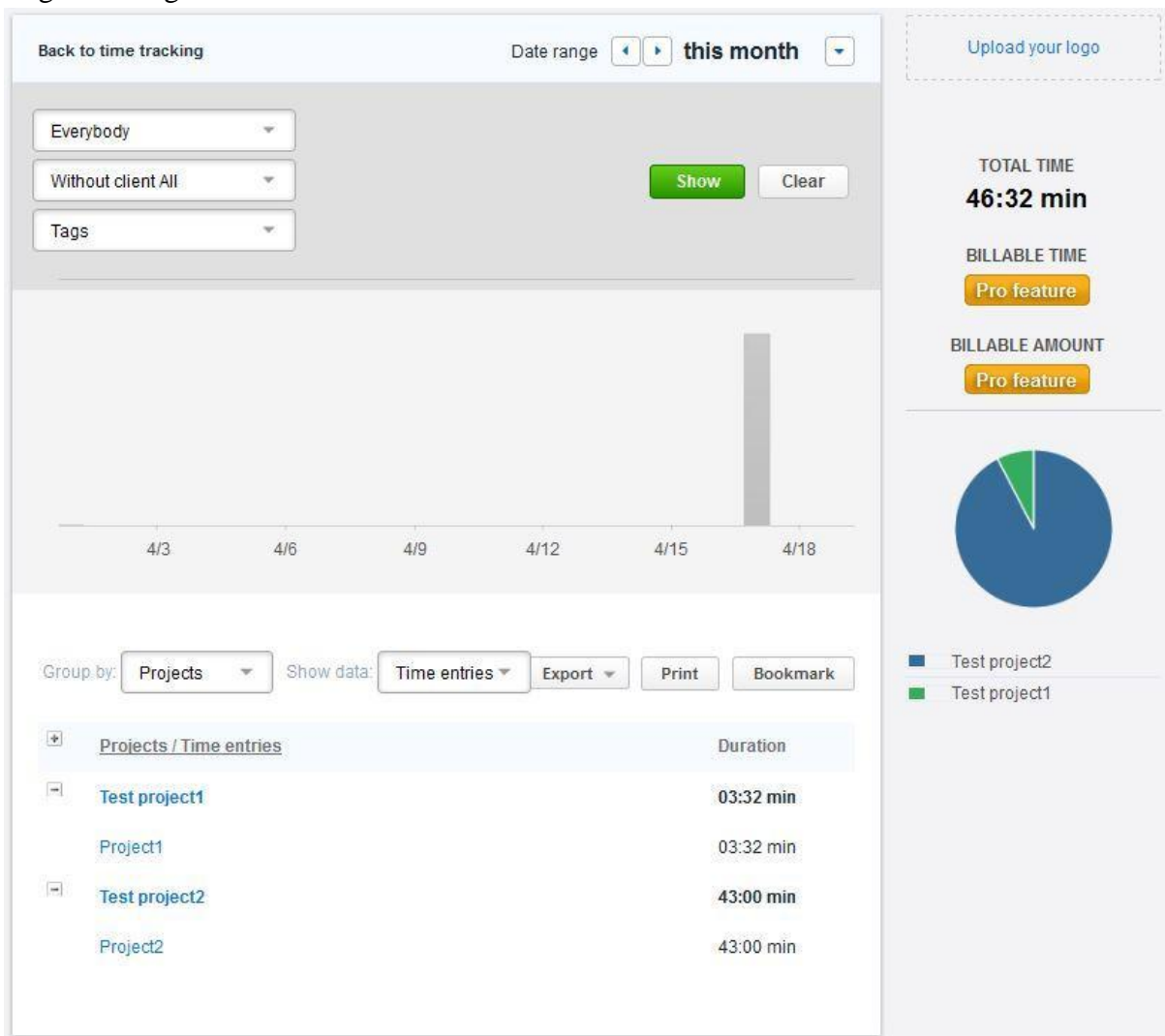
Tartu Ülikool on üks mainekamatest euroopa ülikoolidest, ning selle tunnustatus sõltub peamiselt sellest, kui palju tudengid panustavad õppimisse ning kuivõrd edukad nad on. Selleks, et Tartu Ülikooli hariduse kvaliteedi taset saaks iga aastaga tõsta, on vaja õppejõududel maksimaalselt palju informatsiooni tudengite õpiharjumuste, tulemuste ning eelistuste kohta. Tudengite päevikusüsteemi abil saab ülevaadet tudengite keskmisest koormusest mõne konkreetse aine raames või üldise õppekoormuse kohta mõnes perioodis, samuti saab uurida eelistatud õppimismeetodite kasutatavust ning keskmist tudengite nädalaplaani. Kogutud andmeid aga kasutatakse juba järgmistel aastatel õppimistingimuste paremaks muutmisel.

1.2 Olemasolevad süsteemid

Projekti aluseks olev idee - tudengite õpiharjumuste jälgimise süsteem - on juba Tartu Ülikoolis ellu viidud. Tudengite poolt vormide täitmine, kus nad märgivad õppimisele kulutatud aega, toimub praegu Google Drive teenuse abil, aga tulemuste analüüsi ja graafikute genereerimisega tegeleb R skript. Selline süsteemi ülesehitus ei ole piisavalt mugav ei tudengite ega ka õppejõudude jaoks, seetõttu saigi käesoleva projekti peamiseks eesmärgiks olemasolevale süsteemile mugavama veebilahenduse loomine.

Väljaspool Tartu Ülikooli käesoleva süsteemiga samaväärseid süsteeme autoril leida ei õnnestunud. Samas on palju veebirakendusi ja eraldi programme, mis baseeruvad tudengite õpiharjumuste jälgimise päeviku süsteemiga sarnasel põhimõttel - ajakulu planeerimisel ja jälgimisel. Üheks näiteks on veebist kättesaadav tööriist *Toggl* [1]. *Toggl*'is saab määrata enda või kindla inimeste grupi ajakulu erinevatele tegevustele. Igale tegevusele saab märkida selle algus- ja lõppaeg, kuuluvus mõnesse projekti, ülesande teostaja ning kas on see tasustatav tegevus või mitte. Selleks, et jälgida erinevatele tegevustele kulutatud aega, on *Toggl*'is võimalik vaadata kokkuvõtet kindla perioodi kohta (nagu on näidatud *Joonisel 1*), milles

tuuakse esile projekte ja ülesandeid puudutavaid andmeid koos üldist ajakulu illustreeritavate diagrammidega.



Joonis 1: Ajaaruandluse tulemuste vaatamine Toggl'is

Selle töö raames valmiva päeviku-veebirakenduse üldine põhimõte on sarnane - võimaldada inimestel enda ajakulu jälgida. Peamiseks erinevuseks on see, et uus veebirakendus on ette nähtud ainult Tartu Ülikooli tudengitele kasutamiseks, ning ajakulu saab märkida vaid kindlate ainetega tegelemise, mitte mistahes tegevuste kohta. Samuti on antud päevikusüsteemi üheks iseärasuseks tudengites õppimistahte ergutamine erinevate mänguelementide abil, milledest põhjalikumalt räägitakse teises peatükis.

2. Süsteemi kirjeldus

2.1 Mittefunktsionaalsed nõuded

2.1.1 Kasutatavus

- Veebirakendus peab olema kasutajasõbralik.
- Veateated peavad olema asjakohased.
- Abijuhend peab olema arusaadav ning lihtsasti kättesaadav.

2.1.2 Käideldavus

- Ükski veaolukord ei tohi viia rakenduse kokkuvarisemisele.
- Rakendus peab olema kättesaadav 99% ajast.

2.1.3 Toetus

- Rakendus peab toetama eesti keelset kasutajaliidest.
- Rakendust peab olema võimalik korrektselt kasutada brauserite *Internet Explorer*, *Firefox*, *Opera*, *Chrome* ja *Safari* uuemates versioonides (*Internet Explorer* soovitatavalt versiooniga 8+).

2.2 Funktsionaalsed nõuded

2.2.1 Sisselogimine

Veebilehel *index.jsp* toimub kasutaja sisselogimine. Selleks, et logida sisse, sisestab tudeng või õppejõud enda Tartu Ülikooli kasutajatunnuse ja parooli ning vajutab nuppu *OK* (vt. *Joonis 2*). Kasutades *LDAP* protokoll, autentitakse kasutaja. Kui kasutaja autentimine õnnestus, siis kontrollitakse, kas antud nimega kasutaja on kohalikus andmebaasis olemas: kui on, siis suunatakse ta lehele *form.jsp* (kui kasutaja on tudeng) või *results.jsp* (kui kasutaja on õppejõud), kui ei ole, siis tudengi või õppejõu kasutajanimi koos kasutajanime ja parooli abil genereeritud unikaalse räsiväärtuse ehk koodiga salvestatakse andmebaasi ning suunatakse ümber lehele *user.jsp*. Kui kasutaja autentimine ei õnnestunud, siis tagastatakse veateade koodiga 401, mis tähendab, et serveris tekkis viga autoriseerimisel (*Unauthorized*).

Kasutaja unikaalse koodi genereerimise algoritm on järgmine: võetakse kasutaja paroolist ja nimest koostatud sõne (nt. 'qwerty123juku') ning kasutades *SHA-1* räsifunktsiooni seda krüptitakse. Saadud tulemust salvestatakse kasutaja unikaalse koodina.

Lisafunktsionaalsusena on sisselogimise lehel toodud *Facebook*'i nupud *Like* ja *Send* [6]. Nupule *Like* vajutamisel saab *Graphbook* rakendust märkida meeldivaks kasutades *Facebook*'i poolt pakutavat funktsionaalsust. Vajutades nupule *Send* aga on võimalik saata enda mõne *Facebook* sõbrale sõnumit koos linkiga *Graphbook* pealehele.

Joonis 2: Sisselogimine päeviku süsteemi

2.2.2 Ainele registreerimine

Selleks, et vajalike ainete kohta ajaaruandlust teha, peab esmalt ennast registreerima ainetele. See protsess toimub lehel 'Profiil' (*user.jsp*). Esiteks on vaja valida semester ning seejärel sisestada vastavate ainete koodid plokki 'Ainele registreerimine' (nagu on näidatud Joonisel 3). Aine koodi sisestusväljad on automaattäitmise funktsionaalsusega, seega piisab kolme tähe sisestamisest, et saada nimekirja ainetest, mis sisaldavad enda nimetustes või koodides antud tähed. Aine nimetused on toodud esile automaattäitmisel otsingu lihtsustamiseks, tegelikult on võimalik sisestada üksnes aine kood, kuid sel juhul peab veenduma, et sisestatud kood on õige.

Joonis 3: Ainele registreerimine

2.2.3 Ainele registreerimise tühistamine

Ainetele registreerimise tühistamine toimub lehel 'Profiil' (*user.jsp*) plokis 'Registreeritud ained'. Kui kasutaja ei ole veel ühelegi ainele registreerunud, siis antud plokki ei kuvata.

Et tühistada registreerimist konkreetsele ainele, on esmalt vaja vajutada vastavale aine kirjele ning seejärel nupule 'Kustuta'. Soovi korral on võimalik valida ka mitu ainet (kasutades selleks *ctrl* nuppu või tõmmata ained ümber hiirega) ning kustutada neid ühe korraga (vt. *Joonis 4*). Valitud kirjed on märgitud punase värviga.



Joonis 4: Ainele registreerimise tühistamine

2.2.4 Ajaaruandlus

Ainetele kulutatud aja raporteerimiseks peab tulema lehele 'Ajaaruandlus' (*form.jsp*). Sellel lehel on olemas ajaaruandluse vorm, mis on kujutatud *Joonisel 5*. Siin on, esiteks, vaja valida semester ja kuupäev, mille kohta ajaaruandlust soovitakse teha (vaikeväärtuseks on käesolev semester ja tänane päev). Teiseks, peab määrama aine, selle ainega tegelemiseks kasutatud õppimismeetod ning aeg (ühikuks on tunnid), mida õppetööle kulutati. Kui tudeng raporteerib enda ajakulud ainult osa ainete kohta, siis määratakse ülejäänud ainete jaoks automaatselt õppimisajaks 0 tundi.

Selleks, et aja määramine oleks lihtsam ja kiirem, on ajaaruandluse vormile lisatud funktsionaalsus, mis võimaldab sisestada tundide arvu ning jaotada seda võrdselt valitud ainete vahel. Selle lisavõimaluse kasutamiseks peavad esmalt olema valitud kõik vajalikud ained, siis on lihtsalt vaja kirja panna tundide arvu sisestusvälja 'Tunnid...', mis on nupu 'Jaota' kõrval, ning vajutada nupule 'Jaota'.

Kui tudeng ei tegelenud mõnel konkreetsel päeval ühegi ainega, siis peaks ta ka sellest raporteerima. Selleks on vaja vajutada nupule "Ei õppinud" või määrata iga aine kohta aja väärtuseks 0 tundi - see on sama, mis nimetatud nupule vajutamine.

Vastamata: 89/114

May 2013

Mo	Tu	We	Th	Fr	Sa	Su
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31		

Semester
12/13 kevad

Jaota
Tunnid...

Aine	Meetod	Aeg(h)
Infosüsteemid	Slaidide lugemine	0.5
Õpiprogrammid	Abi küsimine	4.1

Kinnita
Tühista
Ei õppinud

Lisa rida +

Joonis 5: Ajaaruandluse vorm

Kui tudengil on soovi pääseda ajaaruandluse vormile otse ilma sisselogimiseta, siis see on võimalik kasutades isiklikku linki, mida genereeritakse kasutajale vastava unikaalse koodi abil (selle koodi tekitamise detailid on toodud peatükis 2.2.1 Sisselogimine). Antud link on kättesaadav samal lehel 'Ajaaruandlus' (vt. Joonis 6); selle formaat on järgmine:

`< süsteemi host > /Login?key=< kasutaja kood >`

Kui kasutaja läheb selle lingi abil päevikusüsteemi, siis esmalt tehakse päring *Login* servletile, mis otsib kohalikust andmebaasist edastatud koodiga kirjet *User* tabelist, ning kui see õnnestub, siis viib kasutajat ajaaruandluse vormile (lehele *form.jsp*), kui aga antud koodiga kasutajat ei leita, siis suunab sisselogimislehele (*index.jsp*).

Sinu isiklik link ajaaruandlusele pääsemiseks:

<http://ats.cs.ut.ee/graphbook/Login?key=f2d516cf288cfd314a6fe49547c2c605a7bfd78>

Joonis 6: Isiklik link ajaaruandluse vormile pääsemiseks

2.2.5 Raporteerimata päevade vaatamine

Selleks, et tudeng saaks näha, mis päevade kohta tema unustas raporteerida enda ajakulud õppimisele, ajaaruandluse vormil olev kalender (vt. Joonis 5) paneb nende päevade taustaks punast värvi. Kalender näitab ainult viimase 30 päeva jooksul raporteerimata jäänud päevad.

Samuti on võimalik näha ka kogu semestri kohta vastamata päevade koguarv. See informatsioon on kuvatud kalendri kohal (vt. *Joonis 5*). Esimesena näidatakse vastamata päevade arvu, seejärel valitud semestri perioodi päevade koguarvu.

2.2.6 Tulemuste vaatamine

Ainetega tegelemiseks raporteeritud tulemustel põhinevaid andmeid, graafikuid ja diagramme on võimalik vaadata lehel 'Tulemused' (*results.jsp*). Plokis 'Sisesta kriteeriumid' võib määrata ajaaruandluse andmete eristamise alust, ning leida ainult need andmed, mis vastavad sisestatud kriteeriumitele, et kasutada neid graafikute ja diagrammide konstrueerimisel. Määratavad otsimise kriteeriumid:

- väli 'Periood' paneb paigale ajaaruandluste tegemiste ajaperioodi (vaikimisi on määratud viimased seitse päeva);
- väljas 'Aine' saab valida aine, mille kohta otsitavad ajakulu andmed peavad olema raporteeritud;
- kui väljale 'Grupp' on sisestatud grupi identifikaator (või identifikaator koos grupi nimetusega), siis andmete otsimisel võetakse arvesse ainult need ajakulu andmed, mis vastavad toodud grupi liikmetele;
- väljal 'Õppimismeetod' on võimalik kindlaks teha, millist õppimismeetodit peavad sisaldama otsitavad ajaaruandluse andmed.

Nimetatud kriteeriumite sisestamisel saab paika panna kõik kriteeriumid (nagu on tehtud *Joonisel 7*) või jätta mõni rida tühjaks, kusjuures sisestusväljas 'Periood' peab vähemalt üks kuupäev olema määratud.

Sisesta kriteeriumid:

Periood:	<input type="text" value="15.03.2013"/> - <input type="text" value="22.03.2013"/>
Aine:	<input type="text" value="MTAT.03.138 - Tarkvaraprojekt"/>
Grupp:	<input type="text" value="1 - Tarkvaraprojekti liiged"/>
Õppimismeetod:	<input type="text" value="Abi küsimine"/>

Joonis 7: Kriteeriumite sisestamine vastavate tulemuste vaatamiseks


2.2.7 Grupi loomine

Lehel 'Grupid' (*group.jsp*) on võimalik luua uut gruppi, millega edaspidi saavad liituda kõik huvilised. Selleks, et gruppi tekitada, tuleb kirjutada uue grupi nimetus sisestusväljale plokis 'Loo uus grupp' ning vajutada nupule 'Loo' (nagu on näha *Joonisel 8*). Kui sama nimega gruppi veel ei eksisteeri, siis salvestatakse uus grupp andmebaasi ja märgitakse kasutaja selle grupi administraatoriks ja liikmeks, vastasel juhul gruppi ei looda ning kasutajale kuvatakse veateade 'Sama nimega grupp juba eksisteerib!' (vt. *Joonis 9*).

Loo uus grupp

Joonis 8: Uue grupi loomine

Tähelepanu! ×

 Sama nimega grupp juba eksisteerib!

Joonis 9: Veateade sama nimega grupi loomisel

2.2.8 Gruppi sisenemine

Selleks et liituda grupiga, on vaja üle minna lehele 'Grupid' (*group.jsp*), kus sisestada plokis 'Liitu grupiga' olevasse väljasse grupi identifikaatori. Et antud sisestusväljale on lisatud automaattäitmise funktsionaalsus, siis on võimalik otsida vajalikku gruppi ka selle nimetuse järgi, sisestades vähemalt kolm tähte selle nimest ja valides pakutavast nimekirjast nõutud gruppi (vt *Joonis 10*). Seejärel tuleb vajutada nupule 'Liitu'. Kõik kasutajagrupid, kus ta on liige, on kuvatud plokis 'Minu grupid'.

Liitu grupiga

1 - Tarkvaraprojekti liigesed

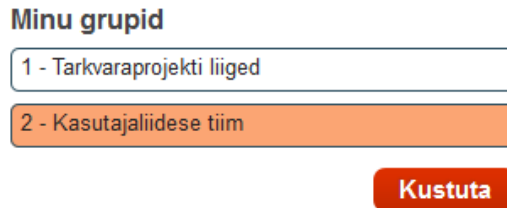
2 - Kasutajaliidese tiim

Joonis 10: Grupiga liitumine

2.2.9 Grupist väljumine

Grupist väljumiseks tuleb lehel 'Grupid' (*group.jsp*) plokis 'Minu grupid' valida kõik need grupid, millede liikmelisust on vaja tühistada, ning vajutada nupule 'Kustuta' (nii kui *Joonisel 11*). Kustutada saab nii üksikühe kui ka mitme kaupa, valides kirjed vajutatud *ctrl* nupuga või tõmmates grupid ümber hiirega. Valitud kirjed on märgitud punase värviga.

Kui kasutaja on administraator mõnes valitud gruppidest, siis grupist väljumisel kustutatakse mitte ainult selle grupi seost kasutajaga, vaid ka gruppi ennast, seega ka teised kasutajad kaotavad antud grupi liikmelisust.



Joonis 11: Grupist väljumine

2.2.10 Välja logimine

Süsteemist välja logida võib igalt siseveebilehelt vajutades nupule 'Logi välja', mis on päises oleva menüü paremas osas. Nupule vajutamisel viiakse kasutaja tagasi sisselogimislehele ning hävitatakse tema sessiooni objekt (vt. *Joonis 12*).

2.2.11 Abijuhendi avamine

Abijuhend on olemas igal süsteemi veebilehel. Selle avamiseks piisab nupule 'Abi' vajutamisest, mis asub lehe päise paremas nurgas nupu 'Logi välja' kõrval (vt. *Joonis 12*). Abijuhendist saab leida informatsiooni kõikide sisestamisväljade ja nuppude kohta, mis asuvad konkreetsetel lehel; see info peaks olema piisav selleks, et edukalt alustada ja lõpetada kõiki tegevusi.



Joonis 12: Nupud 'Logi välja' ja 'Abi'

2.2.12 Tudengi ja õppejõu eristamine

Kui kasutaja logib sisse, siis kohalikust andmebaasist laetakse tema andmeid sisse, ning saadakse väärtust 'is_lecturer', mis määrab, kas sisseloginud kasutaja on tudeng ('is_lecturer=0') või õppejõud ('is_lecturer=1'). Vastavalt sellele kasutajaid eristatakse.

Õppejõul ja tudengil esinevad mõned erinevused rakenduse kasutamise võimalustes. Esimeseks erinevuseks on see, et õppejõud ei näe peamisest menüüst Profiili ja Ajaaruandluse lehtedele viivaid nuppe, sest õppejõu jaoks need lehed pole vajalikud. Teiseks erinevuseks on see, et tudeng ja õppejõud saavad näha erinevad diagrammide komplektid tulemuste lehel - tudeng saab vaadata nii enda kui ka teiste tudengite keskmised tulemused, kusjuures õppejõud saab näha üksnes kõikide tudengite keskmised tulemused (järgmises peatükis tuuakse detailid selle kohta).

2.3 Graafikute ja diagrammide ehitamine

Graafikute komplekti ehitamisel on kasutatud kahte erinevat lähenemist: ühel korral konstrueeritakse graafikuid õppejõu, teisel korral tudengi jaoks. Siin lähtutakse sellest põhimõttest, et õppejõul peaks olema võimalik vaadata kõikide tudengite keskmisi andmeid, kusjuures tudengil endal on tõenäoliselt rohkem huvipakkuv võimalus võrrelda enda tulemusi teiste tudengite keskmiste tulemustega. Iga tudengi privaatsus on võetud arvesse, seetõttu ei

tudeng ega õppejõud ei saa näha konkreetse tudengi tulemusi, ehk üksiku tudengi vastuseid saab näha üksnes tudeng ise.

Mõlema diagrammide ja graafikute komplektide ehitamisel kasutatakse need ajaaruandluste andmed, mis vastavad sisestatud kriteeriumitele (vt. *Joonis 7*).

2.3.1 Diagrammide komplekt õppejõu jaoks

Esimesel real õppejõule kuvatakse valitud perioodi kõikide tudengite keskmine õppetundide arv (vt *Joonis 13*).

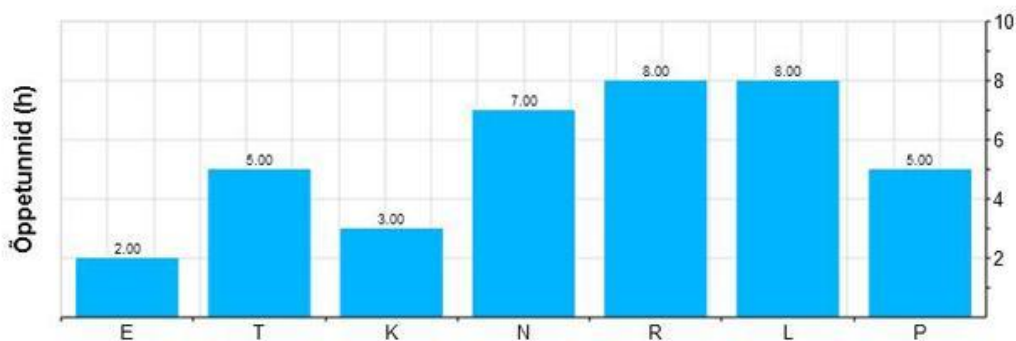
Järgmisena näidatakse tulpdiagrammi, kust saab teada tudengite keskmist õppetöö jaotust nädalapäevade kaupa. See diagramm on väga hea näitaja tudengite õpistrateegiate välja uurimisel.

Kolmandaks diagrammiks on joondiagramm keskmise õppetöö kuvamiseks kuupäevade kaupa; seda näidatakse ainult siis, kui otsingu kriteeriumites perioodiks on valitud rohkem kui 6 päeva ehk perioodiks on vähemalt nädal. Antud kitsendust on sisse viidud selleks, et vältida mõttetut, liiga väikese diagrammi ehitamist. Diagrammi x-teljel on kuvatud kõik kuupäevad, mis on võetud arvesse graafiku konstrueerimisel; kui aga periood on suurem kui 61 päeva, siis näidatakse ainult perioodi algus- ja lõppkuupäev, sest vastasel juhul võib tekkida kuupäevade tekstide ronimine üks teisele.

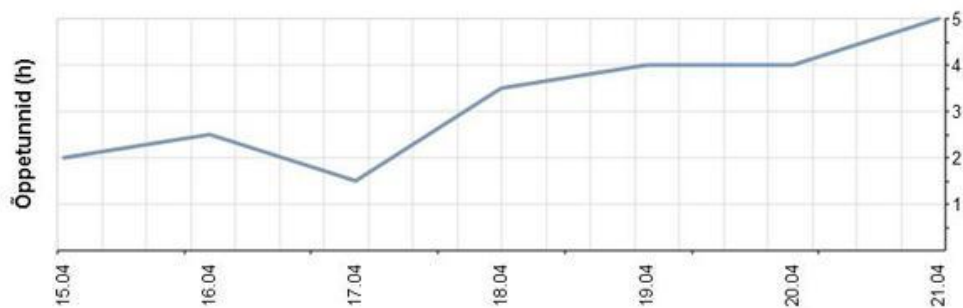
Viimasena joonistatakse sektordiagrammi, kus näidatakse erinevate õppimismeetodite kasutamise protsentuaalse osakaalu. Diagrammi kõrval on olemas legend, kus on paika pandud sektorite ja õppimismeetodite vastavused, ning toodud esile ka saadud protsentide väärtused. Sektorite peale hiirega liikudes, saab teada sama informatsiooni.

Kõikide tudengite keskmine õppetundide arv: 72.0 h

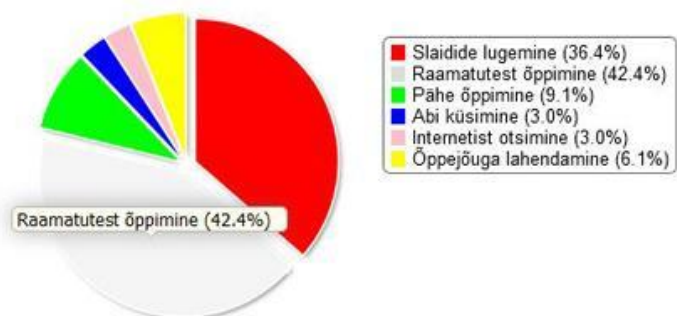
Keskmine iseseisva töö jaotus nädalapäevade kaupa



Keskmine iseseisv töö päevade kaupa



Kõikide tudengite keskmine iseseisva töö jaotus õppimismeetodite järgi



Joonis 13: Õppejõule kuvatavad andmed ja diagrammid

2.3.2 Diagrammide komplekt tudengi jaoks

Juhul, kui otsingu kriteeriumites on määratud konkreetne aine, siis kohe esimesel real tudengile kuvatakse tema vastatud päevade arv ehk valitud aine kohta ajakulude raporteerimise päevade arv määratud perioodis. Kui otsingu kriteeriumina ükski aine ei olnud määratud, siis seda informatsiooni ei kuvata.

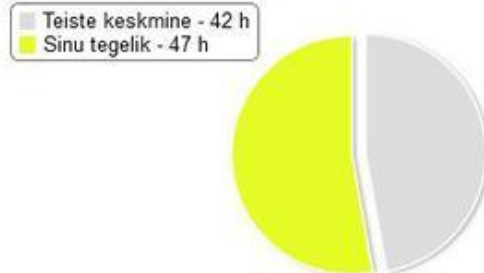
Sektordiagrammil näidatakse teiste tudengite õppetööle kasutatud tundide keskmist ning tudengi enda õppetundide arvu. Siit saab kergesti teada saada, kui palju antud perioodis kulutas aega õppimisele tudeng ise ja kui palju - tema kaaslased keskmiselt.

Järgmised kaks diagrammi on väga sarnased nendega, mis on kuvatavad ka õppejõule: esimesel näidatakse õppetöö jaotust nädalapäevade kaupa, teisel aga keskmisi iseseisva õppetöö tunde kuupäevade järgi - ainuke vahe on see, et tudengile näidatakse nii teiste tudengite andmetel arvutatud keskmisi, kui ka tudengi enda tulemusi, selleks et oleks lihtsam võrrelda enda õppekoormust teiste tudengite omaga (vt *Joonis 14*).

Viimase diagrammina on tulpdiagramm, mis kujutab kõikide tudengite keskmist ning tudengi enda tegelikku õppimismeetodite kasutatavust ainete õppimisel. Tulemusi tuuakse esile protsentides. Antud diagramm on hea abivahend enda ja teiste tudengite õppimisharjumuste võrdlemiseks; näiteks saab otsingu kriteeriumites määrata mõne aine ja vaadata, kui palju ja mis õppimismeetodit kasutades teised õpilased tegelevad valitud ainega, ning vajaduse korral korrigeerida ka enda õpistrateegiat.

Vastatud päevi: 5/7

Teiste tudengite keskmine ja sinu õppetundide arv:



Keskmine iseseisva töö jaotus nädalapäevade kaupa



Keskmine iseseisv töö päevade kaupa



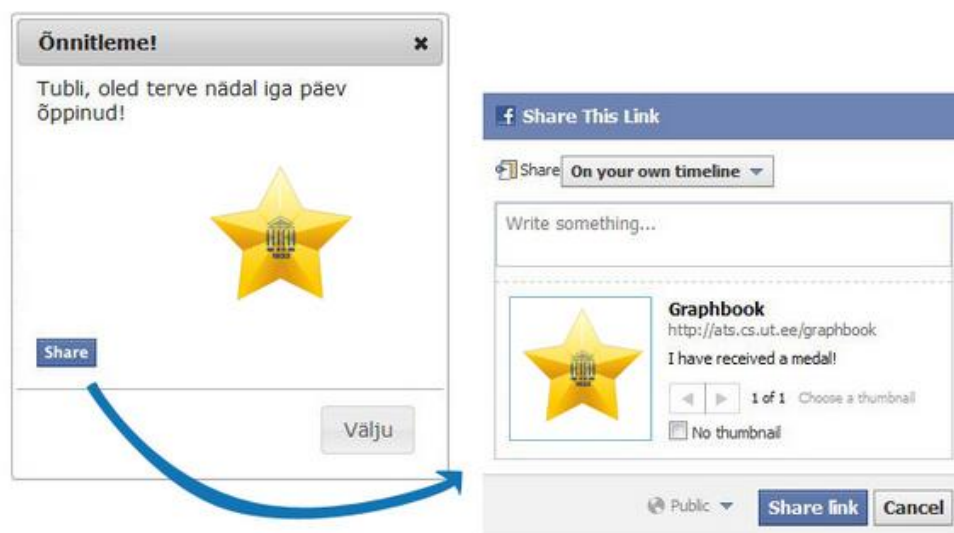
Sinu ja teiste tudengite keskmine iseseisva töö jaotus õppimismeetodite järgi



Joonis 14: Tudengile kuvatavad andmed ja diagrammid

2.4 Medalite saamine

Tudengite õpiharjumuste jälgimise päevikusüsteemile on lisaks õppetöö ajakulu raporteerimisele ja nende andmete alusel ehitatud diagrammide vaatamisele lisatud ka üks lisafunktsionaalsus, mis, projekti autori arvamusel, peaks tudengites ergutama huvi kulutada õppimisega tegelemisele rohkem aega: virtuaalsete medalite välja andmise süsteem. Mida rohkem tudeng õpib ja raporteerib enda ajakulusid, seda suurem on tõenäosus, et ta saab erinevaid medaleid. Medali saamisest on võimalik anda teistele teada läbi *Facebook share* [7] funktsionaalsuse (vt. *Joonis 15*).



Joonis 15: Medali saamise sündmuse jagamine Facebook lehele

Kokku on võimalik saada 6 erinevat tüüpi medalit (*Joonis 16*), nende saamise kriteeriumid on järgmised:

- **Töönädal** - tudeng raporteeris enda ajakulu õppimisele kogu käesoleva nädala jooksul; kui tudeng ei tegelenud ühestki ainetest mõnel päeval, siis ta peaks raporteerima, et ta ei õppinud üldse või määrata iga ainega tegelemise ajakuluks 0 tundi.
- **Töökuu** - tudeng raporteeris enda ajakulu õppimisele kogu käesoleva kuu jooksul; kui tudeng ei tegelenud ühestki ainetest mõnel päeval, siis ta peaks raporteerima, et ta ei õppinud üldse või määrata iga ainega tegelemise ajakuluks 0 tundi.
- **Täistöönädal** - käesoleval nädalal tudeng kulutas rohkem kui 40 tundi õppimisele ja raporteeris sellest päevikusüsteemis.
- **Täistöökuu** - antud kuus tudengil läks õppimisega tegelemisele kokku 160 tundi.
- **Nädala tuupur (täht)** - tudeng raporteeris enda ajakuludest õppimisele kogu käesoleva nädala jooksul ning on sellel nädalal kokku saanud rohkem õppetöö tunde, kui teised tudengid.

- **Kuu tuupur** - tudeng raporteeris enda ajakuludest õppimisele kogu käesoleva kuu jooksul ning on selles kuus kokku saanud rohkem õppetöö tunde, kui teised tudengid.



Joonis 16: Medalite tüübid

3. Kasutatud tehnoloogiad

Autori esimene eelistus oli realiseerida antud veebikeskkonna serveripoolne osa programmeerimiskeeles *Java*. Põhjuseks oli see, et antud programmeerimiskeelt õpetatakse ülikoolis ning seega oli tehnoloogia autorile juba varasemalt tuttav. Selleks et teada saada, kas *Java* on praegusel ajal ikka konstruktiivne valik kaasaegse veebirakenduse valmistamiseks, pöörduti statistika poole. Populaarsuse järgi osutus, et *Java* programmeerimiskeel on endiselt üks enimkasutatavatest keeltest, seetõttu see saigi valituks edasiseks veebirakenduse arendamiseks.

Java keele võimaluste laiendamiseks on võimalik kasutusele võtta mitmeid erinevaid raamistikke (*frameworks*), millest tuntuimad on *Hibernate* ja *Spring*, ja teke (*libraries*), mis läbimõeldud kasutuse korral võivad lihtsamaks teha nii programmeerimise protsessi kui ka edasist arendamist ja koodi muutmist. Kuna antud projekti maht on aga suhteliselt väike, ei osutunud paljude teekide kasutamine otstarbekaks, ja nii piirduti vaid järgmise loeteluga:

- *JSON.simple* [10] teek, mis tegi lihtsamaks *JSON* teksti kodeerimist ja dekodeerimist;
- *Apache Commons* [11] erinevate andmete valideerimise lihtsustamiseks;
- *JDBC API* [12] andmebaasiga suhtlemiseks;
- *UnboundID LDAP API* [13], mis tunduvalt lihtsustab *LDAP* (*LightweightDirectory Access Protocol*) protokolliga autentimise päringu tegemist Tartu Ülikooli serverile;
- *Joda-Time* [14] oli kasutatud kuupäevadega seotud andmetega manipuleerimiseks (nt. kahe päeva vahelise perioodi leidmiseks, kuupäevale vastava nädalapäeva välja uurimiseks jms.).

Veebikeskkonna kliendi poolse osa realiseerimiseks kasutati *JavaScript* koos *JQuery* [5], *JSTL* [8] ja *Ajax* [9] tehnoloogiatega, lisaks sellele ka *HTML5* koos *RGraph* [2] teegiga, mis võimaldab kiiresti ja lihtsasti konstrueerida erinevat tüüpi graafikuid. Andmebaasina sai valitud *MySQL* relatsioonandmebaasi juhtimissüsteem.

3.1 Veebirakenduse serveripoolne osa

3.1.1 Servletid

Serveri poolset osa programmeeriti *Java* keeles, servlettide konteinerina kasutati *Apache Tomcat 6*, mis on vajalik *Java* programmeerimiskeeles kirjutatud veebirakenduse käivitamiseks. Samuti võimaldab antud konteiner *Java* servlettidel (*servlets*) saada kätte kliendi poolsed päringud ning tagastada töödeldud vastused.

Tavaliselt kasutatakse servlette HTTP päringute käsitlemiseks abstraktse klassi *javax.servlet.http.HttpServlet* abil; see sisaldab meetodeid *doGet* ja *doPost*, mis töödeldavad GET ja POST päringud vastavalt. Käesolevas veebirakenduse süsteemis otsustas autor kasutada üksnes POST tüüpi HTTP päringud, sest GET tüüpi päringud ei anna võimalust saata andmeid JSON formaadis, kusjuures see formaat on vajalik enamuses päringutest. Ainuke olukord, millal kasutatakse GET tüüpi päringut, on pääsemine ajaaruandluse isiklikule vormile

ilma sisselogimiseta kasutades ainult kasutajale vastavat koodi, mida genereeritakse kasutaja esimesel sisselogimisel päevikusüsteemi.

3.1.2 Sessiooni objekt, filter ja kuular

Rakendusse sisenemisel luuakse sessiooni objekte, mis edaspidi kasutatakse tudengi või õppejõu tuvastamiseks. Teiste sõnadega, selle abil saab eristada, kas kasutaja on sisseloginud või sisselogimata ning kas ta on tudeng või õppejõud. Selleks salvestatakse sessiooni objekti atribuuti *currentSessionUser*, mille väärtuseks pannakse sisseloginud kasutajale vastav *User* objekt. *User* objektis hoitakse kasutaja identifikaatori, kasutajanime, unikaalset koodi ning staatust *isLecturer*, mis määrab kas ta on õppejõud (*isLecturer=true*) või tudeng (*isLecturer=false*).

Kuular *SessionListener* vastutab sessiooni objektide loomise ja hävitamise eest, samuti määrab see iga sessiooni elueaks 1 tund, see tähendab, et kui kasutaja viibib rakenduse mõnel siseveebilehel aga tunni jooksul ei tee seal midagi, teda logitakse välja automaatselt ning suunatakse ümber sisselogimislehele.

Filter *SessionFilter* on vastutav tühja sessiooni objektiga ehk sisselogimata kasutajate poolt tehtud päringute kinni pidamise eest. Kui autoriseerimata kasutaja proovib rakenduse mõnele siseveebilehele ligi saada, siis *SessionFilter* suunab teda ümber sisselogimislehele. Seega autoriseerimata kasutaja saab viibida ainult sisselogimislehel (*index.jsp*), ning ei ole vahet, mis lehelt ta proovib siseveebilehtedele ligi saada, *SessionFilter* teda sinna ei lase.

3.1.3 Autentimine

Tudengite ja õppejõudude autentimine toimub *LDAP* protokolliga Tartu Ülikooli serverile päringu tegemise kaudu. Ülikooli serverile saadetakse kasutaja poolt sisestatud kasutajanimi ja parool, ning positiivse vastuse (koodi 0) saamisel peetakse autentimist õnnestunuks. Vastasel juhul kuvatakse veateade "Sisselogimine ei õnnestunud!" ja värvitakse kasutajanime ja parooli sisestusväljad punasega.

3.1.4 Kodeeringu filter

Andmete UTF-8 kodeerimisega tulenevate võimalike probleemide lahendamiseks oli loodud filter *CharsetFilter*, mis seab kõikidele sisse- ja väljatulevatele serveri päringutele (*request* ja *response*) kodeerimise tüübiks UTF-8. See tagab eesti keelele omapäraseid sümboleid, nagu "õ", "ä", "ö" ja "ü", sisaldavate andmete edasist korrektset dekodeerimist nii rakenduse serveri kui kliendi poolel.

3.1.5 Andmebaas

Relatsiooniandmebaasi juhtimissüsteemina sai valituks MySQL. Selle sidumiseks *Java* serveriga on võetud kasutusele *JDBC (Java DataBase Connectivity) API*, mis on väga kergekaaluline ja lihtne käsitlemises. Andmebaasiga ühenduse loomiseks oli ainult lisatud MySQL'ile vastav vajalik *mysql-connector-java* draiver, mis asub projekti /WEB-INF/lib kaustas (vt. Lisa 1). Joonis 17 demonstreerib päevikusüsteemi andmemudeli. Andmemudel on kokku 10 tabelit:

- **User** - sisaldab andmeid veebirakenduse kasutajate kohta; kui tudeng logib päevikusüsteemi sisse esimest korda, siis tema kasutajanimi salvestatakse andmebaasi,

kui aga õppejõud tahab süsteemi siseneda, siis pärast esimest sisselogimist peab andmebaasis käsitsi märkida veeru 'is_lecturer' väärtuseks 1, mis tähendab, et antud uus kasutaja on õppejõud. Kasutaja 'is_lecturer' staatuse muutmiseks kasutatakse järgmist päringut:

UPDATE USER SET IS_LECTURER=1 WHERE USERNAME=<kasutajanimi>;

- **Course** - tabel, mis sisaldab kõikvõimalikud ained koos koodidega. Eeldatakse, et süsteemi administraator käsitsi sisestab kõik vajalikud ained andmebaasi enne süsteemi avalikustamist. Selleks kasutatakse järgmist päringut:

INSERT INTO COURSE (CODE,NAME) VALUES (<aine kood>,<aine nimetus>;

- **Semester** - tabel, mis sisaldab kõikvõimalikud semestrite nimetused koos semestrite algus- ja lõppkuupäevadega. Eeldatakse, et süsteemi administraator jooksvalt sisestab andmebaasi uue semestri nimetused ja ajad. Selleks kasutatakse järgmist päringut:

INSERT INTO SEMESTER (TEXT,START_DATE, END_DATE) VALUES (<semestri nimetus>,<semestri alguskuupäev>,<semestri lõppkuupäev>;

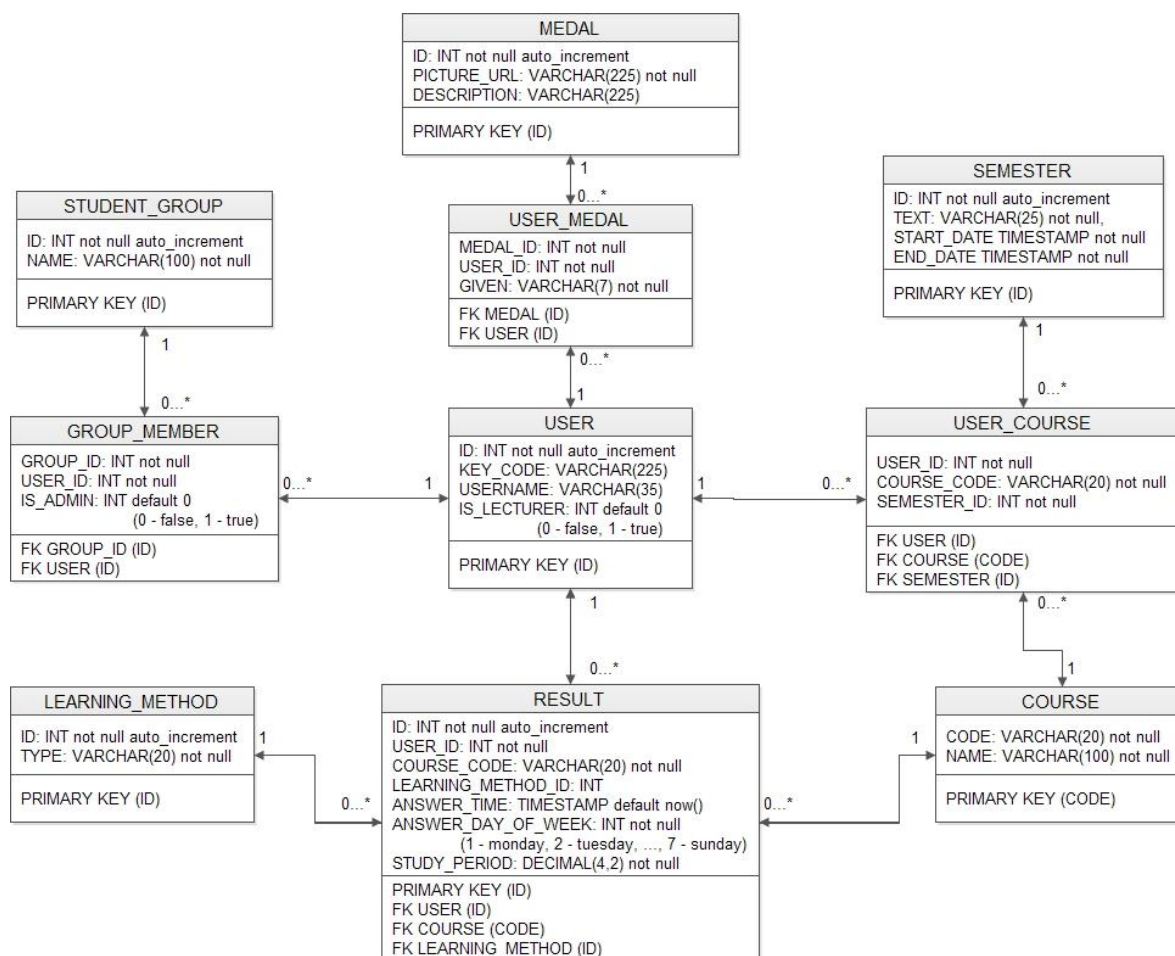
- **User_course** - antud tabelit on vaja kasutaja, ainete ja semestrite sidumiseks, kasutades vastavaid identifikaatoreid tabelitest 'User', 'Course' ja 'Semester'.
- **Learning_method** - tabel, mis sisaldab kõikvõimalikud õppimismeetodite nimetused. Eeldatakse, et süsteemi administraator käsitsi sisestab kõik vajalikud õppimismeetodid andmebaasi enne süsteemi avalikustamist. Selleks kasutatakse järgmist päringut:
INSERT INTO LEARNING_METHOD(TYPE) VALUES (<õppimismeetodi nimetus>;
- **Result** - tabelisse salvestatakse tudengi ajakulu tulemused. Siin märgitakse kindla tudengi identifikaator tabelist 'User', vastava aine kood, õppimismeetodi identifikaator (mida võetakse tabelist 'Learning_method'), kuupäev 'answer_time', mille kohta resultaati salvestatakse, koos sellele kuupäevale vastava nädalapäevaga 'answer_day_of_week' (selle väärtusteks võib olla 1,2,...,7, mis tähendavad vastavalt 'Esmaspäev', 'Teisipäev',..., 'Pühapäev'), ning viimasena veergu 'study_period' salvestatakse ajakulu õppimisele (väärtus antakse tundides).

- **Student_group** - selles tabelis hoitakse loodud gruppide nimed ja identifikaatorid.
- **Group_member** - kasutades antud tabelit, määratakse kasutajate kuuluvust kindlasse gruppi, samuti siin pannakse paigale, kas vaadeldav kasutaja on antud grupi administraator ehk looja (is_admin=1) või mitte (is_admin=0).
- **Medal** - tabel, mis sisaldab andmeid virtuaalsete medalite kohta. Veerg 'picture_url' määrab vastava medali pildi URL'i; veerg 'description' sisaldab teksti, mida näidatakse kasutajale vastava medali saamisel. Eeldatakse, et süsteemi administraator käsitsi sisestab kõik vajalikud medalite andmed andmebaasi enne süsteemi avalikustamist. Selleks kasutatakse järgmist päringut:

INSERT INTO MEDAL(PICTURE_URL,DESCRIPTION) VALUES (<medali pildi url>,<medali kirjeldus>;

- **User_medal** - antud tabelit on vaja kasutaja ja medali sidumiseks, kasutades vastavaid identifikaatoreid tabelitest 'User' ja 'Medal'. Lisaks identifikaatoritele, veerus 'given' määratakse vastava medali saamise perioodi, mille formaadiks võib erinevate medalite

korral olla kas '<aastaNädal>/<aasta>' või '<kuu>/<aasta>' (nt. '16/2013' või '4/2013'). Antud andmete salvestamine aitab vältida olukorda, kus sama medal oleks konkreetsele tudengile mingis perioodis mitu korda välja antud.



Joonis 17: Andmemudel

3.2 Veebirakenduse kliendi poolne osa

3.2.1 JavaScript ja selle teegid

Veebileheküljed on tehtud *HTML (HyperText Markup Language)* ja *CSS (Cascading Style Sheets)* abil. Veebisaidi interaktiivsuse eest vastutab aga skriptikeel *JavaScript* ja selle teegid *jQuery* ja *JSTL*. Nuppude funktsionaalsused, vormide valideerimine, valikmenüüde andmetega täitmine - kõik toimub tänu *JavaScript* keelele. *jQuery* on võetud kasutusele peamiselt kahel järgmisel põhjusel.

Esiteks, teeb see kergemaks rakenduse kasutajasõbralikumaks muutmist. Näiteks, kuupäeva sisestamine tänu *Datepicker* [5] vidinale (Joonis 18) ja gruppide otsimine tekstivälja automaattäitmise (*autocomplete*) funktsionaalsuse (Joonis 10) abil teeb veebirakenduse kasutamist meeldivamaks.



Joonis 18: Kuupäeva sisestamine kasutades Datepicker vidinat

Teiseks, *jQuery* teek lihtsustab *AJAX* (*Asynchronous JavaScript And XML*) tehnoloogia rakendamist, mis võimaldab jooksvat andmevahetust brauseri ja serveri vahel, nii et kasutaja iga nupuvajutamise peale kogu veebilehte uuesti alla ei laetaks. Seda tehnoloogiat kasutatakse käesolevas projektis iga kord, kui on vaja serverisse andmed edastada või saada mõni informatsioon tagasi. Andmevahetus toimub kas eraldi parameetrite saatmise või, suuremahuliste andmete korral, *JSON* (*JavaScript Object Notation*) formaadis andmete listi edastamisega.

JavaScript’i *JSTL*(*JavaServer Pages Standard Tag Library*) teeki kasutatakse peamiselt igale veebilehele vastava päise konstrueerimiseks.

3.2.2 *HTML5* ja graafikute ehitamine

Graafikute ehitamine toimub tänu *HTML5* omasele *canvas* märgendile (*tag*) ja lisateegile *RGraph*, mis toetab erinevate graafikute tüüpide lihtsat konstrueerimist [2].

Käesolevas tudengite päeviku veebirakenduses ehitatakse graafikuid ajakulu tulemuste vaatamise lehel; seal on kasutusel nii tulpdiagrammid, kui ka joon- ja sektordiagrammid (graafikud on esile toodud *Joonisel 13* ja *Joonisel 14*).

Kindlasti on palju ka teisi teke (nt *amCharts*[3] või *Highcharts*[4]), mis oskavad kujutada erinevaid diagramme ja graafikuid, ning tihti ei ole suurt vahet, millist neist kasutada, kuid autor otsustas rakendada *RGraph* poolt pakutavad võimalused, sest see on lihtne ja intuitiivselt arusaadav tehnoloogia. Lisaks sellele on graafikute välimus autori arvates väga esteetiline ning sobilik üldisele veebirakenduse kasutajaliidesele.

4. Kokkuvõte

Antud bakalaureusetöö käigus valminud veebirakendus *Graphbook* baseerub tudengite õpiharjumuste jälgimise päeviku süsteemil. See võimaldab tudengitel raporteerida enda ajakulud ainete kohta, ning saada ülevaadet enda ja teiste tudengite keskmistest, andes ka õppejõududele ligipääsu õpilaste ajakulu keskmistele. Tulemustes tuuakse esile: konkreetse aine kohta raporteeritud ajakulude päevade arvu, õppetööle panustatud tundide arvu, keskmist õppekoormust nädalapäevade ja kuupäevade kaupa ning erinevate õppimismeetodite protsentuaalset kasutatavust. Kõiki tulemusi saab vaadata väljavalitud perioodi, mõne aine, konkreetse õppimismeetodi või valitud grupi liikmete kohta.

Lisaks ajakulu raporteerimisele ja tulemuste vaatamisele on *Graphbook* rakendusse lisatud mõned lisafunktsionaalsused nagu gruppidesse liitumine ja medalite saamine. Gruppidesse liitumine annab võimaluse koguda tudengeid ühte gruppi, sel juhul tulevikus saab tulemuste vaatamisel väga lihtsasti vaadata andmeid konkreetse grupi liikmete kohta. Medalite saamise süsteem aga oli loodud selleks, et ergutada tudengites tahtmist rohkem panustada õppimisse, hakata suhtuma ajaplaneerimisse kui tõsisse oskusesse, mida läheb elus väga tihti vaja.

Veebirakenduse arendamisel tekkis palju erinevaid raskusi. Kõige keerulisemaks osutus autentimata kasutajatele siseveebilehtedele sisenemise keelamine, sest selle funktsionaalsuse ellu viimiseks tuli palju uurida *Java* kuularite kohta. Samuti oli raske tagada arusaadava ja korrektse koodi kirjutamist ning kiire ja mitmefunktsionaalse veebirakenduse loomist. Kõige huvitavamaks projekti osaks oli diagrammide genereerimise võimaluse lisamine, seda eelkõige sellepärast, et diagrammide jaoks andmete kogumise protsess pidi olema lihtsasti arusaadav mitte ainult projekti autorile, vaid ka süsteemi tulevikus edasiarendajatele.

Antud veebirakenduse võimalusi on kindlasti võimalik ka laiendada. Bakalaureusetöö piiratud mahu tõttu ei olnud võimalik panna sinna kõiki lisasid, mis võiksid seda muuta veel rohkem kasutajasõbralikumaks, funktsionaalsemaks ja kiiremaks. Esimeseks võimalikuks ideeks veebirakenduse kiiruse tõstmiseks on näiteks, andmebaasi päringute arvu vähendamine.

Students' study habits observational diary

Bachelor thesis

Julia Gritsuk

Summary

Students' study habits observational diary is a system where students report their study time for each day. This information is then used by students themselves and by teachers to get an overview of students' study plans and habits.

It provides possibility for students to start thinking about allocating their time correctly, about rationally dividing it between many different courses for finishing all tasks before deadlines.

The teachers need this study time information for getting statistics about the actual studying load of students in particular course; this data can also be analyzed by university workers and results of analysis are taken into account when doing any changes in courses' structure or study program.

A similar system is already being used in University of Tartu, but its present implementation is not comfortable to use, thus this project is about developing user-friendly website for mentioned system. The name of the new website was set to *Graphbook*.

In this website users are divided to students and teachers, both can log in into application only with university credentials. Students can register to the courses for particular semester, and then report their spent study time about all or some of the courses for each day. In addition to this, students as well as teachers are able to create groups and join them and see some calculated statistics of all students' reported average study time; whereas students are able to see both their own and other students' average results in order to compare them.

To encourage students to report their study time as often as possible and to pay more attention to time management and study plan, there is an added possibility for students to gain virtual medals. Received medals can be then shared by *Facebook*.

Graphbook system server side is created using *Java* programming language, the client side is written by *HTML*, *CSS* and *JavaScript*, also there are used *jQuery* and *Facebook* libraries. All diagrams are generated by *HTML5* with *RGraph* library.

Viited

Kõik viidatud veebilehed on vaadatud 04.05.2013.

- [1] "Toggl - Insanely simple time tracking", Toggl,
<https://www.toggl.com>
- [2] "RGraph - free HTML5 and JavaScript charts", RGraph JavaScript library,
<http://www.rgraph.net/>
- [3] "amCharts - JavaScript Chart Library", amChart JavaScript library,
<http://www.amcharts.com/>
- [4] "Highcharts - Interactive JavaScript charts for your webpage", Highcharts JavaScript library,
<http://www.highcharts.com/>
- [5] "jQuery UI", jQuery UI,
<http://jqueryui.com/>
- [6] "Like Button - Facebook Developers", Facebook Developers,
<http://developers.facebook.com/docs/reference/plugins/like/>
- [7] "Feed Dialog", Facebook Developers share button,
<http://developers.facebook.com/docs/reference/dialogs/feed/>
- [8] "JSP - Standard Tag Library (JSTL) Tutorial", JSTL,
http://www.tutorialspoint.com/jsp/jsp_standard_tag_library.htm
- [9] "jQuery API Documentation", jQuery.ajax(),
<http://api.jquery.com/jQuery.ajax/>
- [10] "JSON.simple - A simple Java toolkit for JSON", json-simple,
<http://code.google.com/p/json-simple/>
- [11] "Apache Commons", Apache Commons,
<http://commons.apache.org/>
- [12] "Java SE Technologies - Database", JDBC API,
<http://www.oracle.com/technetwork/java/javase/jdbc/index.html>
- [13] "LDAP SDK for Java - UnboundID Solutions", UnboundID LDAP API,
<https://www.unboundid.com/products/ldap-sdk/docs/>
- [14] "Joda-Time - Java date and time API", Joda-Time API for Java,
<http://joda-time.sourceforge.net/>

Lisad

Lisa 1. Veebirakenduse failide sisu lühikirjeldus

Kliendipoolsed failid:

- model.sql - andmebaasi loomiseks vajalik skript koos testandmetega
- index.jsp - sisselogimiseleht, avaleht
- user.jsp - ainetele registreerimise lehe fail (siin kuvatakse ka kasutaja medalid)
- form.jsp - ajaaruandluse tegemise lehe fail
- results.jsp - tulemuste vaatamise lehe fail
- group.jsp - veebilehe fail, kus saab gruppe luua ja grupidesse siseneda
- header.jsp - päise fail, mis kuvatakse iga lehe päises
- footer.jsp - jaluse fail, mis kuvatakse iga lehe jaluses

Serveripoolsed failid:

- dao.ConnectionManager.java - sisaldab andmebaasiga suhtluse loomise meetodit
- dao.*DAO.java - failid, kus tehakse päringud andmebaasi
- enums.DataType.java - *enum*, mida kasutatakse GraphDAO.java failis
- filter.CharsetFilter.java - paneb kõik serveri päringud kasutama UTF-8 kodeeringut
- filter.SessionFilter.java - kontrollib, et autentimata kasutaja ei saaks ligi sisseveebilehtedele
- listener.SessionListener.java - loob ja kustutab sessiooni objektid
- model.*.java - andmebaasi tabelitele vastavad *Java* objektid
- servlet.*.java - kõik servlettid

- service.impl.CloseConnections.java - paneb kinni sidet andmebaasiga
- service.impl.Converter.java - sisaldab m[ajand]ned abimeetodid objektide manipuleerimiseks
- service.impl.JSONDecoder.java - sisaldab meetodid JSON formaadist dekodeerimiseks
- service.impl.JSONEncoder.java - sisaldab meetodid JSON formaati kodeerimiseks
- service.impl.LDAPManager.java - sisaldab meetodit LDAP päringu tegemiseks

- service.impl.graph.GraphConstructor.java - peamine klass diagrammide kuvamiseks vajalike andmete kogumiseks
- service.impl.graph.DayCounter.java - sisaldab meetodit õppepäevade arvutamiseks
- service.impl.graph.HourCounter.java - sisaldab meetodit õppetundide arvutamiseks
- service.impl.graph.LineGraphConstructor.java - siin toimub joondigrammi jaoks vajalike andmete kogumine (õppetundide arvutamine kuupäevade kaupa)
- service.impl.graph.WeekGraphConstructor.java - siin toimub tulpdiagrammi jaoks vajalike andmete kogumine (keskmiste õppetundide arvutamine nädalapäevade kaupa)
- service.impl.graph.MethodGraphConstructor.java - siin toimub erinevate õppimismeetodite protsentuaalse kasutatavuse arvutamine

- `service.impl.medal.MedalBaseCounter.java` - sisaldab meetodit, mis uurib, kas sautaja saab mingit medalit ning vajaduse korral salvestab uue medali saamise fakti andmebaasi
- `service.impl.medal.MedalCalculator.java` - sisaldab meetodeid, mis teevad kindlaks, kas kasutaja saab medalit või mitte

Kaustad:

- `src` - sisaldab kõik serveripoolsed failid kirjutatud *Java* programmeerimiskeeles
- `WEB-INF/lib` - sisaldab kasutatavate teekide failid
- `main` - sisaldab kõik siseveebilehtede *jsp* failid ja allnimetatud kaustad
- `libraries` - sisaldab kõik vajalikud *RGraph* teegi failid
- `additions` - seal asuvad *JavaScript* failid, stiilide fail *css* ja kaust *graphics*
- `graphics` - seal on kõik pildid ja kaust *medals*
- `medals` - seal asuvad medalite pildid

Lisa 2. Rakenduse lähtekood

Graphbook veebirakenduse lähtekood asub antud tööga kaasasoleval CD-plaadil.

Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks

Mina

_____ Julia Gritsuk _____
(*autori nimi*)

(sünnikuupäev: _____ 19.05.1991 _____)

1. annan Tartu Ülikoolile tasuta loa (lihtlitsentsi) enda loodud teose

_____ **Tudengite õpiharjumuste jälgimise päevik** _____
_____,
(*lõputöö pealkiri*)

mille _____ juhendaja _____ on
_____ Margus Niitsoo _____,
(*juhendaja nimi*)

1.1.reprodutseerimiseks säilitamise ja üldsusele kättesaadavaks tegemise eesmärgil, sealhulgas digitaalarhiivi DSpace-is lisamise eesmärgil kuni autoriõiguse kehtivuse tähtaja lõppemiseni;

1.2.üldsusele kättesaadavaks tegemiseks Tartu Ülikooli veebikeskkonna kaudu, sealhulgas digitaalarhiivi DSpace'i kaudu kuni autoriõiguse kehtivuse tähtaja lõppemiseni.

2. olen teadlik, et punktis 1 nimetatud õigused jäävad alles ka autorile.

3. kinnitan, et lihtlitsentsi andmisega ei rikuta teiste isikute intellektuaalomandi ega isikuandmete kaitse seadusest tulenevaid õigusi.

Tartus, ____13.05.2013____ (*kuupäev*)